International Journal of Advanced Robotic Systems

OPEN ACCESS
ARTICLE

Hybrid Architecture for Coordination of AGVs in FMS

Regular Paper

Eduardo G. Hernandez-Martinez[1,*], Sergio A. Foyo-Valdes[2],
Erika S. Puga-Velazquez[2] and Jesús A. Meda-Campaña[2]

1 Engineering Department, Universidad Iberoamericana, Lomas de Santa Fe, Álvaro Obregón, México City, Mexico
2 Mechanical Engineering Department, Posgraduate and Research Section, ESIME Zacatenco, Instituto Politécnico Nacional, Mexico City, Mexico
* Corresponding author E-mail: eghm2@yahoo.com.mx

Received 08 May 2013; Accepted 20 Dec 2013

DOI: 10.5772/57572

Abstract This paper presents a hybrid control architecture that coordinates the motion of groups of automated guided vehicles in flexible manufacturing systems. The high-level control is based on a Petri net model, using the industrial standard ISA-95, obtaining a task-based coordination of equipment and storage considering process restrictions, logical precedences, shared resources and the assignment of robots to move workpieces individually or in subgroups. On the other hand, in the low-level control, three basic control laws are designed for unicycle-type robots in order to achieve desired formation patterns and marching behaviours, avoiding inter-robot collisions. The control scheme combines the task assignment for the robots obtained from the discrete-event model and the implementation of formation and marching continuous control laws applied to the motion of the mobile robots. The hybrid architecture is implemented and validated for the case of a flexible manufacturing system and four mobile robots using a virtual reality platform.

Keywords Petri Nets, Multi-robot Systems, Formation Control, Marching Control, Unicycles

1. Introduction

Currently, some production systems require the coordination of automated Flexible Manufacturing Systems (FMS) for the assembly of different and concurrent products [1]. Petri net (PN) formalism have been widely applied to modelling and supervisory control for FMS [2, 3]. The structure and properties of a PN can represent the asynchronous actions, blocking, concurrency and other dynamic behaviours that frequently appear in large FMS [4, 5].

The Material-Handling System (MHS) is crucial for the productivity of an FMS. This transports the workpieces between storage locations and workstations, and is composed commonly of conveyor belts or manipulators mounted in rails. This kind of MHS becomes fixed and non-reconfigurable transport setup, which could be inadequate for highly flexible manufacturing systems, where new routes could appear in some product sequences. To increase the flexibility of the MHS, several studies have recently appeared in the control systems community on the implementation of groups of mobile robots or Automated Guided Vehicles (AGVs), emulating the coordinated work achieved by a group of workers. An example of groups of robots in a manufacturing industry is the Kiva system [6, 7], where mobile storage locations are charged and moved by small autonomous robots; the human operators remain standing while the products come to them. In [8], the Kiva system includes information about inventories and work orders. The collective behaviour of groups AGVs presents some advantages such as redundancy and fault tolerance when a robot is broken, and the loading of large objects by subgroups of robots in specific formation patterns [9].

A coordinated control of AGVs in an FMS must enable group behaviours like formation control, path following maintaining a formation pattern (marching), and collision avoidance between robots or static obstacles [10]. Also, this coordination must obey process specifications, fault-tolerance strategies and the assignment of routes according to some rules of distances or equitable energy distribution [11]. In terms of the continuous dynamic behaviour of multi-robot systems, the coordination of AGVs has been mentioned by the control systems community as a possible application of the generic problems of consensus, formation control, marching control, collision avoidance and dispersion [12]. The control schemes encompass behaviour-based control laws [13], swarm stability [14], virtual structures [15], leader-follower schemes [16], and artificial potential functions [17]. The case of coordination of unicycle-type robots to achieve formation and marching behaviours has been widely studied by the robotics community due to the drawbacks of non-holonomic restrictions. For example, distributed control laws with limited information for the group tracking of a predefined path are proposed in [18], motion coordination control algorithms for convergence to a desired geometric pattern using are given in [19], stabilization of distributed local tracking is analysed in [20], and a biology-inspired decentralized navigation law with limited information about other robots in the group is designed in [21].

Despite the importance of analysis of the performance of control laws for point robots and unicycle-type robots, no previous work has clarified how these low-level control strategies can be applied to the case of manufacturing systems, or their interconnection to a coordination layer of tasks for robots within a production system. A few works, like [22], propose some hybrid architectures of formation control and a planning level, in this case neural networks, for dispersion tasks. Despite the potentiality of multi-robot mobile systems in FMS, mobile robots researchers have focused only on the design of generic control laws related to the formation, marching and collision avoidance of groups of robots in experimental platforms, assuming that these behaviours can be adopted in an industrial context, which must operate under the requirements of industrial standards such as ISA-95 [23]. On the other hand, the discrete-event community has studied only the high-level behaviour of FMS, excluding the analysis of the motion of the AGVs [24, 25]. There are few recent studies on the combination of discrete-event models and multi-agent robot systems. For example, in [26] a PN model coordinates some agents in computer systems while avoiding deadlock.A new type of agent-based PN is defined in [27, 28] for the communication of computer systems. Finally, a supervisory control for Finite State Automata using AGVs is obtained for an FMS in [29], where the drawback of state explosion is presented in a real scenario.

This paper proposes a hybrid architecture that models an FMS using PN in a high-level coordination. The PN model represents the concurrency of tasks, the logic of precedence between tasks, the limitation of storage and the availability of robots. In the low-level coordination, the control selects adequate AGVs to transport the pieces and implements continuous control laws to achieve formations, marching behaviours and collision avoidance of groups of robots. The approach clarifies the overlap between the two control levels, facilitating implementation in engineering practice.

The paper is organized as follows. Section 2 presents the problem statement. Section 3 summarizes the main concepts of PN. Section 4 presents the modelling framework of PN for the FMS. Section 5 defines the formation, marching and collision avoidance control laws for the AGVs. Section 6 presents the case study of four AGVs moving in an FMS. Finally, Section 7 presents some concluding remarks.

2. Problem statement

A general scheme of an FMS is presented in Figure 1 (see [1]), composed of a set of raw material storage locations, which provides workpieces for a set of machines, commonly Computer Numerical Control (CNC) machines, which can be programmed to perform different machining of the same raw material. The manufactured pieces of every machine are moved to specific slots in intermediated storage. Commonly, the intermediated storage is installed in matrix-shaped storage locations, where each row contains the parts manufactured by the same machine. When a product is required, the adequate number and type of parts contained in intermediate storage are transported to the assembly stations where different assembly programmes are carried out. The final assembled products are moved to final product storage and the FMS becomes available for other subsequent processes. A set of homogeneous AGVs are added to the system to transport the raw material, machined parts or final assembly products between the different equipment of the FMS. Note that, in a general case, one or many workpieces could be transported by one or many AGVs, enabling formations of robots in specific geometric patterns and group following of routes in the workspace. As shown in Figure 1, the AGVs are initially in a home position, where the robots can be connected to a battery charging station. When a robot finishes a task, it can return to its home position or attend to another transportation task.

The main objective is to design control architecture that coordinates the AGVs and the process tasks, achieving maximum concurrency and obeying process restrictions in a general FMS. The PN must achieve process task ordering, equipment and storage availability representation and assignment of AGVs for the transportation of workpieces. On the other hand, the robots must receive the necessary information to implement formation, marching and collision avoidance control laws for group transportation of pieces. The next section describes the basic PN principles that need to be observed to analyse the general modelling of an FMS with AGVs.

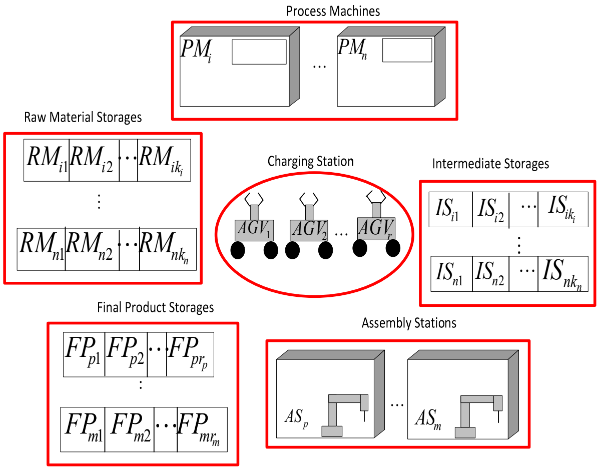


Figure 1. General scheme of an FMS

3. Petri Net basic definitions

Definition 1: According to [2, 3], a PN with finite capacity is a weighted and bipartite graph given by a 5-tuple

$$PN = (P, T, F, W, M_0), \qquad (1)$$

where:

- $P = \{p_1, p_2, \dots, p_m\}$ and $T = \{t_1, t_2, \dots, t_n\}$ are the disjoint sets of nodes called places and transitions, respectively.
- $F \subseteq (P \times T) \cup (T \times P)$ is the set of arcs, connecting places to transitions and vice versa, with elements $w(p,t), w(t,p), \forall t \in T, \forall p \in P$.
- W:F→ Z^+ is the function that assigns the weights to each arc.
- $M(p): P \to Z^+$, represents an m-entry vector with the number of tokens residing inside each place. Let M_0 be the initial marking, and consider that each$p_i \in P$ has a finite capacity $\mathrm{c_i}$ of tokens; then, $M(p_i) \leq c_i$.

The reachability $R(PN)$ is the set of all possible markings reachable from $\mathrm{M_0}$. A k-th state or marking in a PN, denoted by $\mathrm{M_k}$ is achieved according to the following transition rule.

Definition 2: A transition $\mathrm{t_i} \in \mathrm{T}$ is said to be enabled in a PN with finite capacity if:

- $M_k(p_j) \geq w(p_j, t_i), \forall p_j | w(p_j, t_i) \in F$, with j=1,…,m and i=1,…,n.
- $w(t_i, p_j) + M_k(p_j) \leq c_j, \forall p_j \in P$.

If $\mathrm{t_i}$ is enabled in the *k-th* firing in some firing sequence, then the next marking is defined by

$$\mathrm{M_{k+1}(p_j) = M_k(p_j) - w(p_j, t_i) + w(t_i, p_j)} \qquad (2)$$

4. PN modelling framework for the FMS

4.1 Definition of storage locations, machines, assembly stations and process tasks

According to the problem statement presented in Section 2, consider $\mathrm{PM = \{PM_1, \dots, PM_n\}}$ as the set of process machines (PM), $\mathrm{AS = \{AS_1, \dots, AS_m\}}$ as the set of assembly stations (AS) and $\mathrm{AGV = \{AGV_1, \dots, AGV_r\}}$ as the set of AGVs. It is assumed that each $\mathrm{PM_i, i = 1, \dots, n}$ performs $\mathrm{M_{i1}, \dots, M_{ik_i}}$ machining programmes. Every machining $\mathrm{M_{ij}}$ with $\mathrm{i = 1, \dots, n, j = 1, \dots, k_i}$ requires the raw material contained in the raw material storage locations (RM) $\mathrm{RM_{ij}}$. The $\mathrm{RM_{ij}}$ is manually loaded and has a capacity limited to $\mathrm{C(RM_{ij})}$ pieces. When the machining $\mathrm{M_{ij}}$ has finished, the manufactured part is allocated to an intermediated storage location, numbered as $\mathrm{IS_{ij}}$ with a capacity limited to $\mathrm{C(IS_{ij})}$. Denoted by $\mathrm{A_1(RM_{ij}, PM_i, n_{ij})}$ is the AGVs' task of transporting a raw material from $\mathrm{RM_{ij}}$ to $\mathrm{PM_i}$ using $\mathrm{n_{ij} \leq r}$ AGVs to perform the machining $\mathrm{M_{ij}}$ and denote by $\mathrm{A_2(PM_i, IS_{ij}, m_{ij})}$ is the task of transporting the manufactured parts of the machining program $\mathrm{M_{ij}}$ from $\mathrm{PM_i}$ to $\mathrm{IS_{ij}}$ using $\mathrm{m_{ij} \leq r}$ mobile robots. Note that $\mathrm{m_{ij} \leq n_{ij}}$ because the machining $\mathrm{M_{ij}}$ generally removes material from the workpiece, decreasing the dimension and weight of the manufactured part. If $\mathrm{n_{ij}, m_{ij} > 1}$, this implies that the robots achieve a formation control and path following that will be detailed in the next section.

Each assembly station $\mathrm{AS_p, p = 1, \dots, m}$ performs $\mathrm{S_{i1}, \dots, S_{ir_p}}$ assembly programs. Each $\mathrm{S_{pq}}$ with $\mathrm{p = 1, \dots, m, q = 1, \dots, r_p}$ assembles the parts allocated to the different intermediate storage locations. Let $U(S_{pq}) \subseteq \{IS_{11}, \ldots, IS_{nk_n}\}$ be the set of intermediate storage locations that contains the parts needed to complete the assembly program S_{pq} and $K(S_{pq}, IS_{ij}) \in Z^+, \forall IS_{ij} \in U(S_{pq})$ the quantity of parts of IS_{ij} to achieve the assembly program S_{pq}. Then, we define $A_3\left(IS_{ij}, AS_p, m_{ij}\right)$ as the task related to the transportation of a part from IS_{ij} to AS_p to perform the assembly program S_{pq} using the same m_{ij} number of AGVs. Note that the previous task must be repeated $K(S_{pq}, IS_{ij})$ times, but not necessarily by the same robots. Finally, when the assembly program S_{pq} has finished, then the final assembly product is placed in the final product storage location FP_{pq}, which has a capacity limited to $C(FP_{pq})$, and is manually unloaded. Denoted by $A_4(AS_p, FP_{pq}, t_{pq})$, the transportation of the final assembly product made by S_{pq} from AS_p to FP_{pq} now uses $t_{pq} \leq r$ robots. Figures 1 and 2 show schematically the storage locations, PM, AS and their process tasks, and the routes of the AGVs in the system, where the labels of some arrows are related to the number of pieces $K(S_{pq}, IS_{ij})$ moved by the robots. Note that A_1, A_2, A_3 and A_4 are the four kinds of transportation tasks realized by the AGVs.

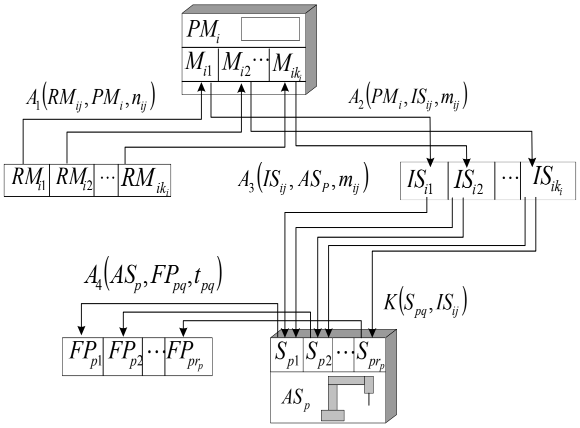

PM_i
M_{i1} M_{i2} ⋯ M_{ik_i}
$A_1(RM_{ij}, PM_i, n_{ij})$
$A_2(PM_i, IS_{ij}, m_{ij})$
RM_{i1} RM_{i2} ⋯ RM_{ik_i}
$A_3(IS_{ij}, AS_p, m_{ij})$
IS_{i1} IS_{i2} ⋯ IS_{ik_i}
$A_4(AS_p, FP_{pq}, t_{pq})$
$K(S_{pq}, IS_{ij})$
FP_{p1} FP_{p2} ⋯ FP_{pr_p}
S_{p1} S_{p2} ⋯ S_{pr_p}
AS_p

Figure 2. Scheme with process task

The PM, AS and AGVs can be classified using the reference model in the ISA-95 standard [23], according to Figure 3, where a clear separation of the MHS (the AGVs) and the process workstations (machines and assembly stations) is proposed. Then, the process tasks are assigned to each equipment device, illustrated at the bottom of Figure 3. ISA-95 proposes that product sequences be reduced to the correct order of the process tasks (product recipe) commonly addressed in a computer-based system, avoiding the reprogramming of the routines in the local controllers of the FMS elements when a new product is required.

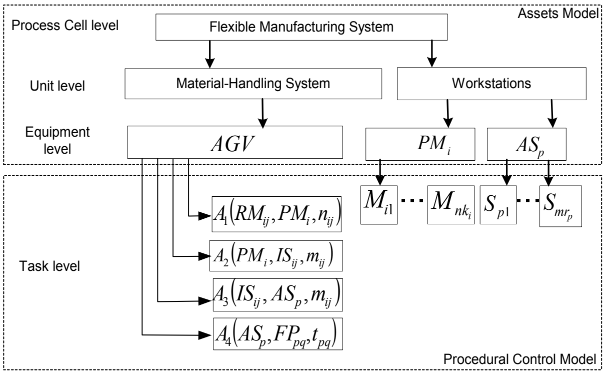


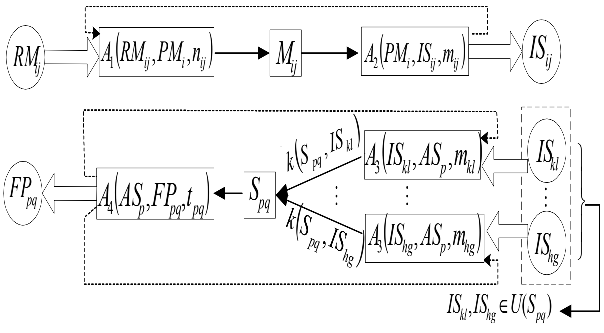
Figure 3. Equipment and task decomposition using the ISA-95 standard

Using the notation and terminology defined previously, some generic models of PN are constructed in order to obtain a general model of the FMS.

4.2 PN model of the system

Figure 4 presents a graphic representation, based on a precedence diagram, of the relationships between tasks (rectangles) and storage locations (circles). Note that each wide arrow symbolizes a relation between a storage location and material handling tasks performed by the AGVs, the continuous arrows are direct precedences and the dotted arrows represent inverse precedences between a pair of tasks.

The diagram in Figure 4 is divided into two sections. The first corresponds to the manufacture of parts. The AGVs transport the raw material from the storage RM_{ij} to the machine PM_i (for the machining task M_{ij}) through the task A_1; when the machine finishes, the AGVs transport the machined subpart from the machine PM_i to intermediate storage through the task A_2. The direct logical precedences between $A_1 - PM_i$ and $PM_i - A_2$ establish that only when the precedent task ends does the subsequent task start. When the AGVs finish the task A_2, they can once again transport raw materials to the machine PM_i, avoiding conflict at the entrance of the machine. There are "inverse" logical precedences (dotted line) between $A_2 - A_1$, explained in detail in the next subsection. The second section of Figure 4 corresponds to the assembly operations. The subparts are transported from the storage locations IS_{ij} to the assembly station AS_p using the task A_3. Recalling the weight $K(S_{pq}, IS_{ij})$ establishes the number of necessary parts of the storage IS_{ij} to complete an assembly S_{pq}. When S_{pq} is finished, the final product is transported by the AGVs to storage FP_{pq} through a task A_4. Finally, there is an inverse dependence of $A_4 - A_3$, which ensures that the AGVs reload the assembly station only when the previous product has been stored in FP_{pq}.



RM_{ij}
$A_1(RM_{ij}, PM_i, n_{ij})$
M_{ij}
$A_2(PM_i, IS_{ij}, m_{ij})$
IS_{ij}
FP_{pq}
$A_4(AS_p, FP_{pq}, t_{pq})$
S_{pq}
$k(S_{pq}, IS_{kl})$
$A_3(IS_{kl}, AS_p, m_{kl})$
IS_{kl}
$k(S_{pq}, IS_{hg})$
$A_3(IS_{hg}, AS_p, m_{hg})$
IS_{hg}
$IS_{kl}, IS_{hg} \in U(S_{pq})$

Figure 4. Logical precedences in a general FMS

4.3 Basic PN models

The relations between the storage locations and tasks, as presented in Figure 4, can be reduced to (1) the process tasks, (2) unloading or loading at a storage location, (3) the logical precedence (direct or inverse) between a pair of tasks, and (4) the conditions of gathering parts for an assembly and the starting of a new assembly. These relationships are examined below.

4.3.1 Tasks of AGVs, machines and assembly stations

Models of the tasks performed by the AGVs, machines and assembly stations are shown in Figure 5. The prefixes "s" and "f" denote the start and finish of tasks, respectively. The four types of AGV tasks ($A_i, i = 1, \dots, 4$) shown in Figure 5a can be summarized as $A_i(X, Y, w)$, where X denotes the place where the robots pick up the pieces, Y represents the place where the robots drop these pieces, and w is the quantity of robots needed to perform the task. The value of w is related to the type of transportation according to the table in Figure 5a. The number of AGVs is translated to the tokens in the place named AGV. Note that when a task $A_i(X, Y, w)$ starts, the w tokens are removed from AGV and returned when the task finished. The selection of the AGVs assigned to each task could be performed according to the smallest distance of the robots with respect to the point where the robots pick up the piece.

The machining and assembly tasks are represented in Figures 5b and 5c, respectively. Note that the places PM_i, AS_p contain one token, denoting that every machine or assembly station can carry out one process task at the same time.

4.3.2 Storage models

The storage models in the FMS are classified in three types: (a) manual load-automatic unload, as in dispensers of raw material (RM_{ij}) like entries of the FMS, (b) automatic load-automatic unload, for intermediate storage locations of subparts (IS_{ij}), and (c) automatic load-manual unload, which involves the storage of final products (FP_{pq}) when the products leave the FMS. The three kinds of storage models and their translation to PN are shown in Figure 6. In Figure 6a, the load of storage RM_{ij} requires a manual input transition IN_{ij} that puts tokens at RM_{ij}, which are extracted by the start of a task $T_a \in A_1$. In Figure 6b, storage IS_{ij} is loaded by the end of a task $T_{a.} \in A_2$ and unloaded by the start of tasks T_b ... $T_c \in A_3$. Finally, Figure 6c shows storage locations loaded by the end of a task $T_a \in A_4$ (final product) and unloaded by a manual transition OUT_{pq}that absorbs tokens from the place FP_{pq}.

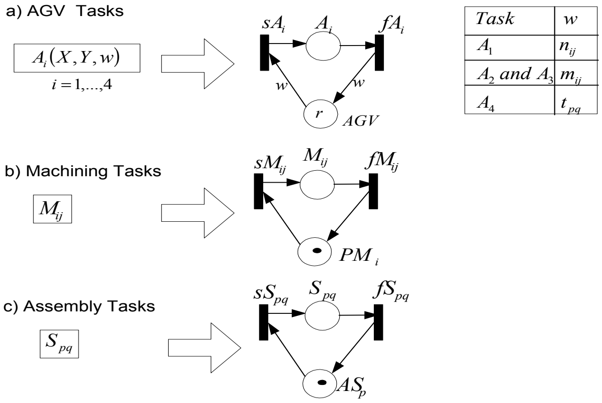


Figure 5. Models of tasks and translation PN

4.3.3 Logical precedences between tasks

The logical precedences represent the correct functional flow of the tasks during the FMS execution. There exist two basic kinds of precedences: direct logical precedences (D-direct) and inverse logical precedences (D-inverse).

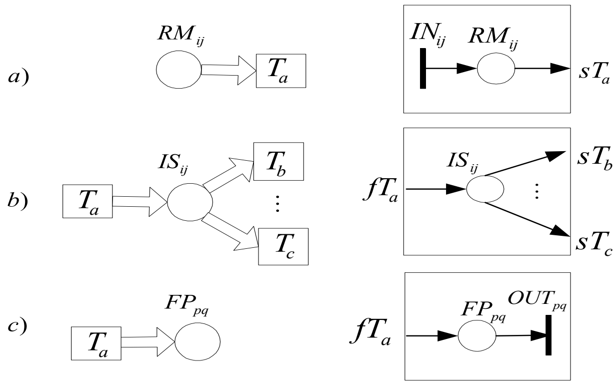


Figure 6. Storage models and translation PN

A D-direct appears when, in the normal functional flow of the FMS operation, the finish of a previous task enables the beginning of a subsequent task. Figure 7a shows a simple graphical representation of a D-direct between tasks T_a and T_b, where the boxes are tasks (left side = start and right side = end). The PN translation of the D-direct is given in Figure 7b. Note that the initial marking is equal to zero.

A D-inverse occurs when the finish of a posterior task enables the start of an initial task, in a normal functional flow of the FMS, for example when the end of a machining task enables the feeding of a new part. Figure 8a shows a D-inverse, where the continuous line has been changed to a dotted line. Now, the end of the subsequent T_b enables the start of $T_{a.}$ The translation of a D-inverse to PN is shown in Figure 8b. Note that the tokens at the place of the D-inverse are not zero, because it is necessary to enable the beginning of the task T_a at the start of the functional flow.

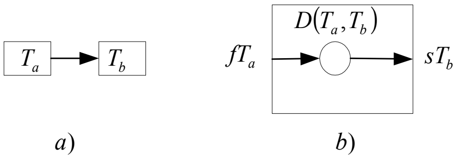


Figure 7. D-direct precedence models

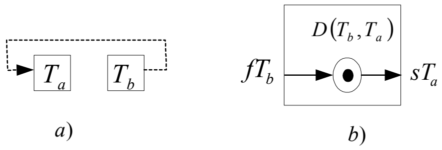


Figure 8. D-inverse precedence models

4.3.4 Logical precedences with multiple tasks

Extending the simple models of D-direct and D-inverse, we obtain general models for the conjunction of multiple tasks that enable the start of multiple posterior tasks. It appears for example in the case of transportation of subparts to enable the starting of an assembly task. Figures 7 and 8 can be extended to encompass the models in Figure 9a, showing the synchronization of two or more different tasks T_a ... $T_b \in A_3$ required to achieve the start of an assembly task $T_c \in S_{pq}$. Note that $K(T_c, IS_{ij})$ is a weight added to each output arc denoting the number of subparts in IS_{ij} needed to start the task S_{pq}. Figure 9b shows an inverse dependence, where the end of task $T_d \in A_4$ allows the start of two or more tasks T_a ... $T_b \in A_3$ for a new assembly. The PN translation needs the same quantity of places D as tasks T_a ... $T_b \in A_3$; each place initially contains $K(T_c, IS_{ij})$ tokens representing the amount of necessary subparts for IS_{ij} for each assembly S_{pq}; the input arcs to the places D also have the weight $K(T_c, IS_{ij})$.

Applying the PN models of AGVs, machines, assembly stations, storage and logical precedences to the main diagram of the FMS shown in Figure 4, the result is a general discrete-event model with maximum concurrency of tasks. The case study shows in detail the translation of the PN models.

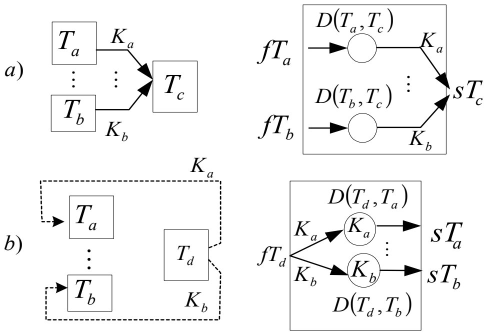


Figure 9. Logical precedences for the conjunction of different tasks

Remark 1: The modelling approach is a novel use of PN, based on the industry standard ISA-95. The rules of the PN evolution will serve as a high-level coordinator for the equipment and the mobile robots. The main contribution is to provide clarity for the construction of generic blocks and the interconnection to generate complex PN. This focus facilitates the study of real industrial FMS and its implementation in supervisory systems.

5. Coordination of the AGVs

In the previous section, the discrete-event dynamics of the FMS were modelled as a PN, where the task of motion of the AGVs is related to the quantity of AGVs that transport workpieces between machines, assembly stations and storage locations of the FMS. Thus, at a low level some motion control laws must be designed to achieve formations and to follow prescribed trajectories in the FMS. A brief introduction about the standard formation and marching control [30] is presented in the next subsection.

5.1 Kinematic models of AGVs

Denoted by $N = \{R_1, \ldots, R_n\}$, a subset of n unicycles are moving in the plane. Note that $n \leq r$, where r is the quantity of AGVs available in the FMS. The kinematic model of each agent or robot R_i, as shown in Figure 10, is described by

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \underbrace{\begin{bmatrix} cos\theta_i & 0 \\ sin\theta_i & 0 \\ 0 & 1 \end{bmatrix}}_{R_i(\theta_i)} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}, i = 1, \ldots, n \qquad (3)$$

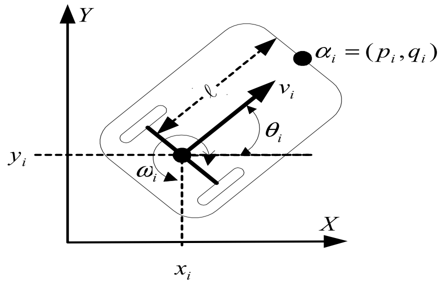


Figure 10. Kinematic model of unicycle-type robots

where $\mathrm{v_i}$ is the linear velocity of the midpoint of the wheels axis, $\mathrm{\omega_i}$ is its angular velocity and $R_i(\theta_i)$ the rotation matrix. In this work, we consider the position of the front point $\mathrm{\alpha_i = (p_i, q_i)}$, shown in Figure 10 as the objective of control. This coordinate can be the centre of mass of the vehicle or the place where a sensor or actuator is placed. Also, the previous condition avoids singularities in the control law and facilitates the analysis simplification in the case of a point robot, as mentioned below [31, 32]. The coordinates α_i and their dynamics are given by

$$\alpha_i = \begin{bmatrix} p_i \\ q_i \end{bmatrix} = \begin{bmatrix} x_i + \ell\cos(\theta_i) \\ y_i + \ell\sin(\theta_i) \end{bmatrix}, i = 1, \dots, n \qquad (4)$$

$$\dot{\alpha}_i = \underbrace{\begin{bmatrix} \cos\theta_i & -\ell\sin\theta_i \\ \sin\theta_i & \ell\cos\theta_i \end{bmatrix}}_{A_i(\theta_i)} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}, i = 1, \dots, n \qquad (5)$$

where $A_i(\theta_i)$ is the so-called decoupling matrix of every $\mathrm{R_i}$, which is non-singular because $\mathrm{det(A_i(\theta_i)) = l \neq 0}$. Then, it is possible to design a control strategy for the positioning of $\mathrm{\alpha_i}$ at a desired location using the control law

$$[v_i, \omega_i]^T = A_i^{-1}(\theta_i) f_i, i = 1, \dots, n \qquad (6)$$

where f_i is the desired dynamics of coordinates $\mathrm{\alpha_i}$. Note that in the closed-loop system (5)-(6), the dynamics of α_i are reduced to $\mathrm{\dot{\alpha}_i = f_i}$, and the behaviour of the front point is similar to the case of a simple point robot moving in the plane, cancelling all the non-linearities of the original kinematic model. Note that the control law (6) does not influence directly the orientation angles $\mathrm{\theta_i}$, but these could converge, as mentioned in [30]. The main objective of the control law is to design the functions $\mathrm{f_i}$ to achieve (1) formation, (2) convergence to a point, and (3) marching behaviour, avoiding inter-robot collisions in the three cases.

5.2 Formation control and collision avoidance

The main aim of formation control is to achieve a formation pattern defined by the relative positions of every robot with respect to team members. Let $\mathrm{N_i \subseteq \{\alpha_1, \dots, \alpha_n\}}$, $\mathrm{N_i \neq \emptyset, i = 1, \dots, n}$ denote the subset of positions of the robots which are detectable for $\mathrm{R_i}$. If $\mathrm{\alpha_j \in N_i}$, then a relative position vector $\mathrm{c_{ji} = [h_{ji}, v_{ji}]^T}$ is defined which represents the desired position of $\mathrm{\alpha_i}$ with respect to $\mathrm{\alpha_j}$ in a particular formation. For a general case, the desired position of $\mathrm{\alpha_i}$ with respect to all the detectable robots can be expressed by

$$\begin{aligned} \alpha_i^* &= \frac{1}{n_i} \sum_{j \in N_i} (\alpha_j + c_{ji}), i = 1, \dots, n-1 \\ \alpha_n^* &= \frac{1}{n_n + 1} \left(\sum_{j \in N_n} (\alpha_j + c_{jn}) + \tau \right) \end{aligned} \qquad (7)$$

where $\mathrm{n_i}$ is the cardinality of $\mathrm{N_i}$. Thus, the desired relative position of $\mathrm{R_i}$ can be considered as a combination of the desired positions of $\mathrm{\alpha_i}$ with respect to the positions of all elements of $\mathrm{N_i}$. Note that $\mathrm{\alpha_n^*}$ includes $\mathrm{\tau \in R^2}$, which denotes a reference point within the work area known by the $\mathrm{n-th}$ robot only (assigned as leader robot). The existence of a reference point is necessary to converge to a specific area of the workspace. It is clear that the position of τ depends on the place where the storage locations and machines in the FMS are established.

The main objective is to design a formation control law for every robot $\mathrm{R_i}$, such that $\mathrm{\lim_{t\to\infty}(\alpha_i - \alpha_i^*) = 0, i = 1, \dots, n}$ (convergence to the desired formation) and $\mathrm{\|\alpha_i(t) - \alpha_j(t)\|} > d, \mathrm{\forall t \geq 0, i \neq j}$ (collision avoidance) where d is the diameter of a circle centred on the coordinate α_i that circumscribes each robot. According to [25], a formation control law with inter-robot collision avoidance is given by

$$f_i = -\tfrac{1}{2}k\left(\tfrac{\partial\gamma_i}{\partial\alpha_i}\right) - \tfrac{1}{2}\eta\left(\tfrac{\partial V_i}{\partial\alpha_i}\right), i = 1, \dots, n \qquad (8)$$

where $k, \eta > 0$ $\gamma_i = \|\alpha_i - \alpha_i^*\|^2$ is a (positive) attractive potential function with $\mathrm{\gamma_i = 0}$ only when $\mathrm{\alpha_i = \alpha_i^*}$. The function V_i is the repulsive potential function, proposed by Khatib [26], defined between a pair of robots that violate the minimum allowed distance given by $\mathrm{V_i = \sum_{j \in M_i}\left(\frac{1}{\|\alpha_i - \alpha_j\|^2} - \frac{1}{d^2}\right) i = 1, \dots, n,}$ where $\mathrm{M_i = \{\alpha_j \mid \|\alpha_i - \alpha_j\| \leq d\}, i = 1, \dots, n}$. Note that every robot $\mathrm{R_i}$ always senses the position of the robots that belong to $\mathrm{N_i}$, and eventually they can sense the closer robots using proximity sensors. The control law (8) describes an artificial vector field where the robots are attracted to the desired position and eventually avoid inter-robot collisions.

5.3 Convergence to a point in the plane

When a robot $\mathrm{R_i}$ requires only to converge to a static point $\mathrm{\beta_i \in R^2}$, a modification of the control law (8) can be given by

$$\tilde{f}_i = -\tfrac{1}{2}k\left(\tfrac{\partial\rho_i}{\partial\alpha_i}\right) - \tfrac{1}{2}\eta\left(\tfrac{\partial V_i}{\partial\alpha_i}\right), i = 1, \dots, n \qquad (9)$$

where $\rho_i = \|\alpha_i - \beta_i\|^2$. The location of the point $\mathrm{\beta_i}$, as mentioned below, could coincide with the home position of the robot, which includes a battery charge station.

5.4 Marching control

Assuming that the robots are formed in a desired pattern, [33] proposes the next marching control strategy, where the leader follows a desired marching path $\mathrm{m(t)}$ and the follower robots maintain a rigid formation in relation to the leader. This marching control law is given by

$$\overline{f}_i = -\frac{1}{2}k\left(\frac{\partial \gamma_i}{\partial \alpha_i}\right) - \frac{1}{2}\eta\left(\frac{\partial V_i}{\partial \alpha_i}\right) + \dot{m}(t), i = 1, \dots, n \qquad (10)$$

$$\overline{f}_n = \dot{m}(t) - k_m(\alpha_n - m(t))$$

where $k_m > 0$ is a gain parameter. Note that the derivative of the marching path must be communicated to all the followers to ensure that the formation errors converge to zero as shows in [33]. The control law (10) includes formation, group path following and collision avoidance.

5.5 Example of a transportation task

To illustrate the use of the control laws (8)-(10), suppose that the robots must realize a transportation task $A_1(RM_{12}, PM_1, 3)$, i.e., three robots work together to move a piece from the raw material storage location RM_{12} at the workspace coordinate $[-60,30]$ to the process machine PM_1 located at the coordinate $[0,70]$. Assume that $N_1 = \{\alpha_2\}$, $N_2 = \{\alpha_3\}$, $N_3 = \{\alpha_1\}$, i.e., the robots communicate to form a cyclic pursuit configuration [30], and they need to achieve a triangle-shaped formation pattern given by $c_{21} = [0,11], c_{32} = [5.5, -5.5], c_{13} = [-5.5, -5.5]$. Figures 11, 12 and 13 show a numerical simulation of the three robots carrying the task $A_1(RM_{12}, PM_1, 3)$ with $k = 0.2, k_m = 100, \eta = 1 \times 10^7, d = 5$ and $l = 1$.

The simulation shows three intervals of time where the three control laws (8)-(10) are implemented. The initial positions of the four robots are $\beta_1 = (-10.5,0)$, $\beta_2 = (-3.5,0)$, $\beta_3 = (3.5,0)$ and $\beta_4 = (10.5,0)$.

In Figure 12, for $0 \leq t < 250$, three robots are moved from their given home positions to the position of RM_{12} in the triangle formation using the formation control law with collision avoidance (8). Note that the robots R_1, R_2, R_3 are chosen due to the minor distance with respect to the position of RM_{12}, and R_3 is selected as the leader. Note that the formation errors defined by $e_i = \alpha_i - \alpha_i^*, i = 1, \dots, 4$ converge to zero, where e_{ix} and e_{iy} refer to the error of robot i on the x and y axes, respectively. The position and orientation of the robots in the time instants $t = 15s$ and $t = 250s$ are depicted in Figure 11.

For $250 \leq t \leq 500$, the formed robots apply the marching control law (10), where the marching path is given by the parametric equations for a straight line that begins in the position of RM_{12} and ends in the position of PM_1. Note in Figure 12 that the error coordinates remains at zero, i.e., the formation is rigid during the path following. Figure 11 shows the marching of the robots in the time instants $t = 375s$ and $t = 500s$.

Finally, in Figure 12, for $500 < t \leq 750$, the robots have finished the transportation of the workpiece and they use the control law (9), breaking formation to return to their home positions, avoiding again inter-robot collision. The coordination errors also converge to zero. Figure 11 shows the posture of the robots in the time instants $t = 505s$ and $t = 511s$, very close of the home positions.

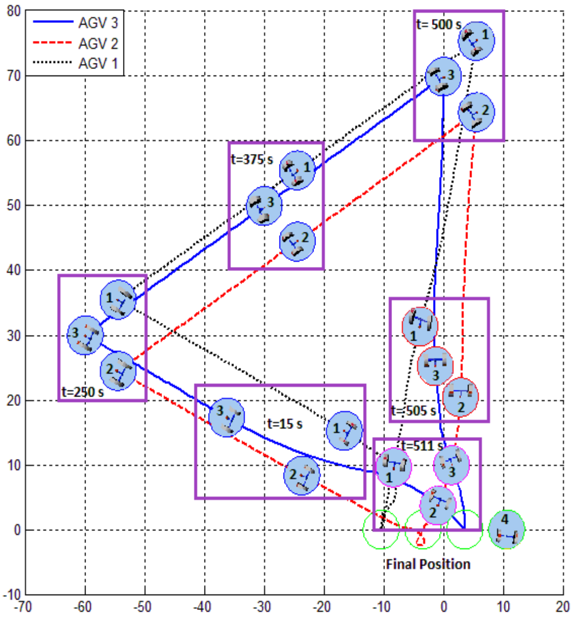


Figure 11. Implementation of the motion control laws

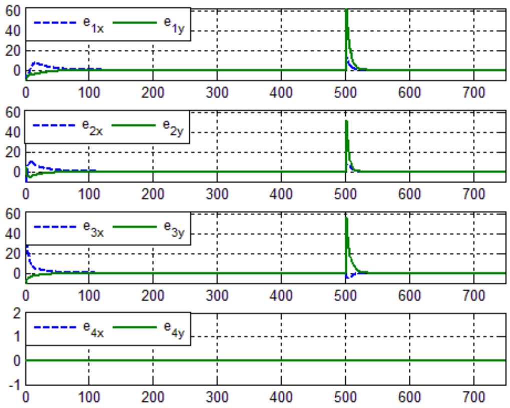


Figure 12. Graphic of the error coordinates of the robots

Figure 13 shows that during the three time intervals the variables d_{ij}, which denote the distance between robots R_i and R_j, are greater than the diameter d, i.e., the robots do not collide. The time interval $250 \leq t \leq 500$ shows that the distance between robots satisfies the proposed pattern formation. Finally, in $500 < t \leq 750$ the robots eventually break the formation and return to home. Note that robot 4 remains at home because it is not used by the task. On the other hand, if the robots have finished the transportation of a piece, they can be assigned to another task instead of implementing control law (9) and returning to their home position.

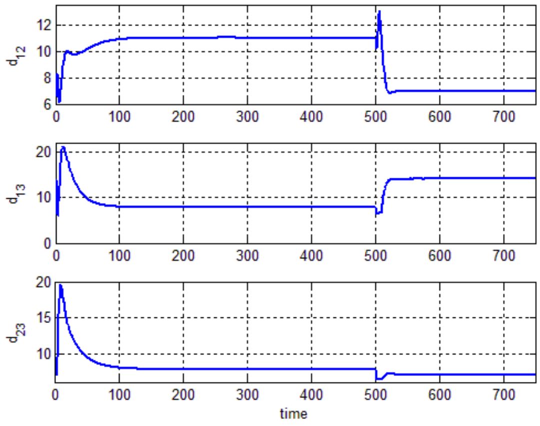


Figure 13. Distance between AGVs

Remark 2. Note that the coordination of the AGVs in the approach is composed of two levels. In the higher level, the PN model enables the transportation tasks considering the availability of the robots and the restrictions of the process. The AGVs are selected according to the shortest distance to the initial point of the task. The PN model has the capacity to execute concurrently different transportation tasks, as shown in the example in the next section. On the other hand, every task in low-level control implements continuous control laws for the robots to achieve the desired motion behaviour. Therefore, the hybrid architecture solves the problems of task assignment and convergence to a formation, tracking and collision avoidance at the same time.

6. Case study

Figure 14 shows the FMS including a set of $r = 4$ AGVs, two PM (PM_1 and PM_2) and two AS (AS_1, AS_2). PM_1 performs $k_1 = 2$ machining programs M_{11}, M_{12} and PM_2 performs $k_2 = 1$ machining program M_{21}. Thus, the raw material storage locations are RM_{11}, RM_{12} or RM_{21}.

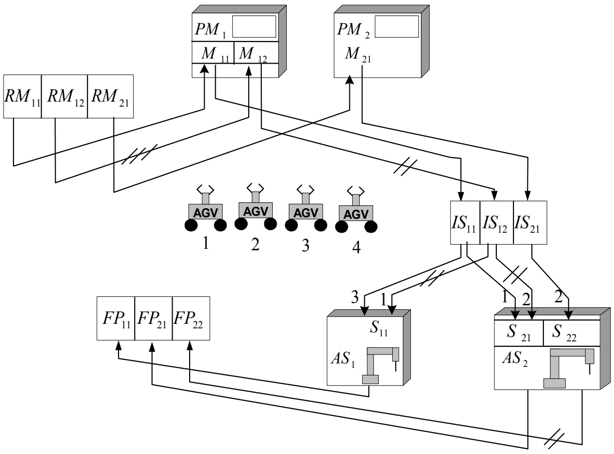


Figure 14. Scheme of the elements and tasks of an FMS

For the transportation of the raw material RM_{12} to PM_1, it is necessary that $n_{12} = 3$ AGVs (denoted by little crossed lines). Note that in the scheme of Figure 14, RM_{11} and RM_{21} only require one AGV (by default omitting the line cross). The AGVs move the machined parts to intermediate storage at IS_{11}, IS_{12} or IS_{21}. Observe that two AGVs $m_{12} = 2$ are needed to move the parts from PM_1 to IS_{12}.

For the case of the assembly stations, AS_1 performs $r_1 = 1$ assembly program S_{11} and AS_2 performs $r_2 = 2$ assembly programs S_{21}, S_{22}. The necessary subparts to achieve the products are defined by $U(S_{11}) = \{IS_{11}, IS_{12}\}, U(S_{21}) = \{IS_{11}, IS_{12}\}$ and $U(S_{22}) = \{IS_{21}\}$ in the quantities given by $K(S_{11}, IS_{11}) = 3$, $K(S_{11}, IS_{12}) = 1$, $K(S_{21}, IS_{11}) = 1$, $K(S_{21}, IS_{12}) = 2$ and $K(S_{22}, IS_{21}) = 2$. Note that two AGVs are required to move pieces from IS_{12} to AS_1 or AS_2.

The final product storage locations are FP_{11}, FP_{21}, requiring one AGV; FP_{22} requires two AGVs for transportation. The RM, IS and FP storage locations have capacity $C(RM_{ij}) = 4$, $C(IS_{ij}) = 5$ and $C(FP_{pq}) = 3$, respectively.

According to the general model of logical precedences, Figure 15 shows the diagram of this case study. Using the models presented in section 4, the PN of the FMS is shown in Figure 16.

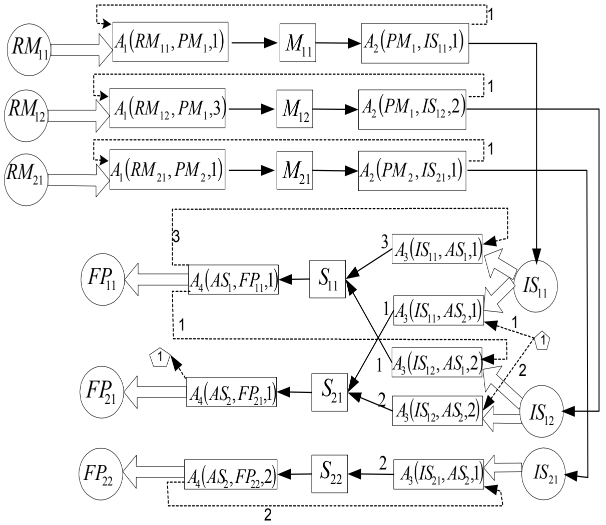


Figure 15. Diagram of precedences of the FMS

Remark 3. Note that the PN approach represents the concurrence that appears in an FMS in a more compact form than another planners like the automata approach. However, for a large FMS the PN model becomes less compact, especially if there exist many restrictions in the process. Despite this, it is possible to identify the generic blocks described in section 4.2, which construct the complete net. Therefore, the final PN model is a systematic method to achieve complex models based on

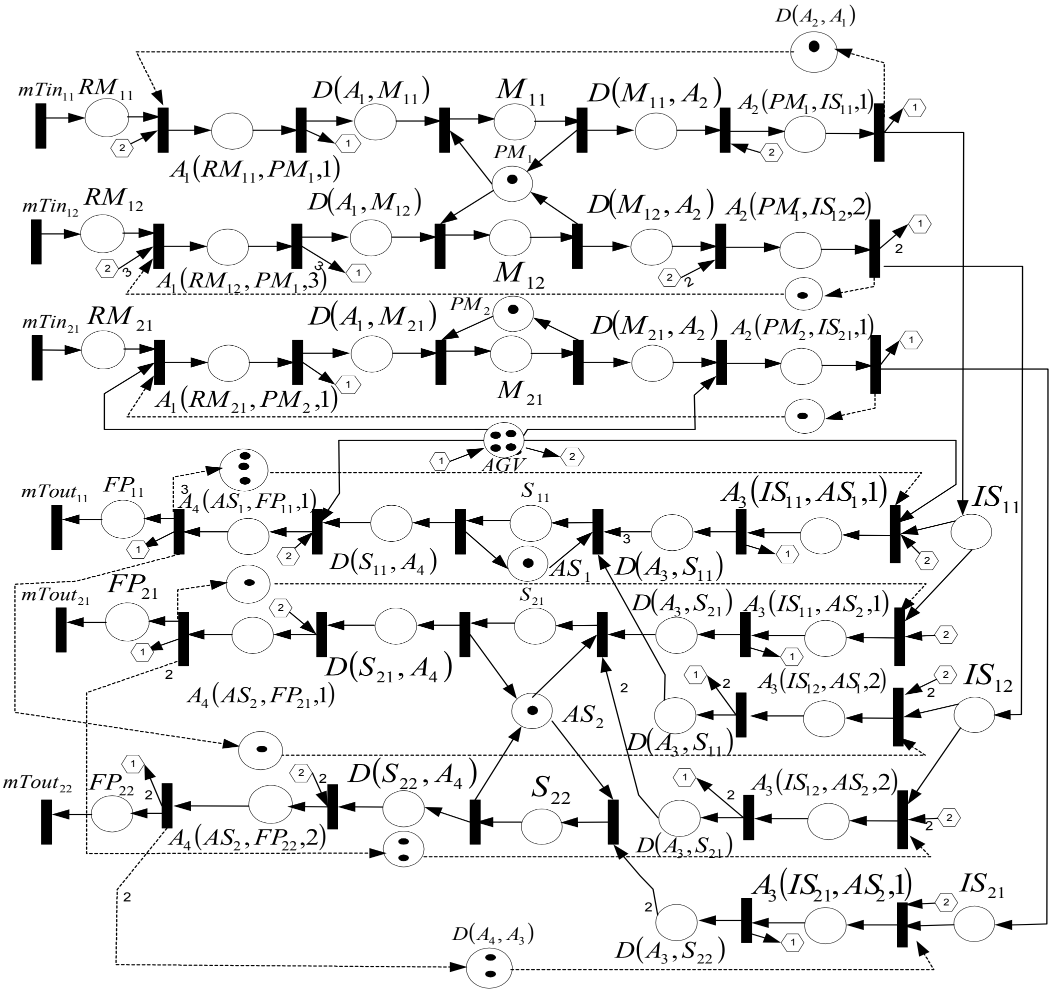


Figure 16. PN model of the FMS case study

simple blocks. On the other hand, the graphic representation of the PN model serves only as a visual tool in the process flow. In a real implementation, the control implementation will be realized using the mathematical structure instead of the graphical interpretation. Thus, the modelling approach based on generic blocks facilitates the translation to real supervisory control systems using the simple rules of the simple PN approach. In this sense, other PN formalism, like coloured PN, fuzzy PN, and others could complicate the codification of the PN and the definition of more properties of the generic blocks described in section 4.2.

Using a simulator of the PN dynamics, a phase-space diagram of the tasks of AGVs, PM and AS is given in the Figure 17. Note that the transportation tasks A_1, A_2, A_3 and A_4 of AGVs include the quantity of AGVs in each task.

To show the task distribution of the AGVs in the FMS, observe that the four robots are occupied in different routes, with task concurrency and distribution of the AGVs at the time instant t_α (see Figure 17).

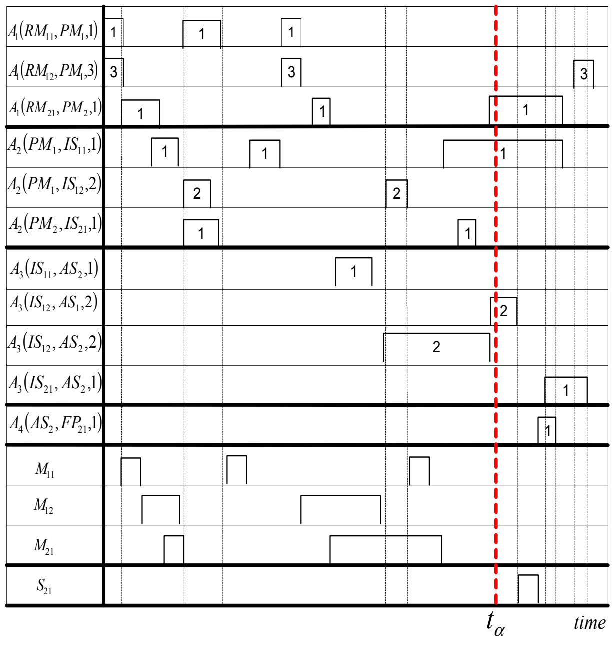


Figure 17. Phase-space diagram of distribution of AGV in concurrent tasks

7. Conclusions

This paper has presented a methodology for the discrete-event modelling of FMS based on the task decomposition proposed by the industrial standard ISA-95 and its translation to generic PN models. Interconnected equipment, storage locations, process tasks and inter-task logical precedences comprise a general PN model that describes the maximal concurrency obeying the process restrictions. The modelling approach allows the modification of the amount of equipment or storage limitations without necessitating changes to the network topology, preserving its static properties. The task-based modelling also provides a clear separation between the generic process tasks related to the equipment capacities and their interconnections for the manufacturing of different and concurrent products. The PN model provides the dynamical structure for a high-level supervision of the FMS in a control computer and the vigilance of the availability of equipment. The coordination of the AGVs is achieved by the implementation of formation, collision avoidance and marching control laws where the robots collaborate in concurrent transportation tasks. The approach adopts a systematic method that uses the concepts of discrete-event systems in an engineering manufacturing context, and clarifies the application of continuous motion control laws of AGVs in FMS.

8. Acknowledgements

This work was partially supported by the Universidad Iberoamericana, CONACyT (SNI scholarship) and IPN through research project 20140659 and scholarships EDI, COFAA, and PIFI.

9. References

[1] Groover S. M. P (2008), Automation, production systems and computer integrated manufacturing, Prentice Hall New Jersey USA.

[2] Cassandras C. G, Lafortune S (2008), Introduction to discrete event systems, Kluwer Academic, New York, USA.

[3] Murata T (1989), Petri nets: Properties, analysis and applications, Proc. IEEE, vol. 77, no. 4, pp. 541-580.

[4] Zhuo M. C, Dicesare F (1993), Petri net synthesis for discrete event control of manufacturing systems, Boston, MA, Kluwer.

[5] Castillo I, Smith J.S (2002), Formal modelling methodologies for control of manufacturing cells: Survey and comparison, Journal of Manufacturing Systems, vol. 21, no. 1, pp. 40-57.

[6] D'Andrea R, Wurman P (2008), Future challenges of coordinating hundreds of autonomous vehicles in distribution facilities, Conference on Technologies for Practical Robot Applications, TePRA. IEEE International, pp. 80-83.

[7] Wurman P. R, D'Andrea R, Mountz M (2008), Coordinating hundreds of cooperative, autonomous vehicles in warehouses, AI Magazine, vol. 29, No. 1, pp. 9-20.

[8] Enright J. J, Wurman P. R (2011), Optimization and coordinated autonomy in mobile fulfillment systems, Automated Action Planning for Autonomous Mobile Robots, AAAI, pp. 33-38.

[9] Xingyan L, Parker L.E (2007), Sensor analysis for fault detection in tightly-coupled multi-robot team tasks , International Conference on Robotics and Automation, pp. 3269-3276.

[10] Ailon A, Zohar I (2012), Control strategies for driving a group of nonholonomic kinematic mobile robots in formation along a time-parameterized path, IEEE/ASME Transactions on Mechatronics, vol. 17, pp. 326-336.

[11] Fan Y, Fei L, Shirong L, Chaoliang Z (2010), Hybrid formation control of multiple mobile robots with obstacle avoidance, 8th World Congress on Intelligent Control and Automation (WCICA), pp. 1039-1044.

[12] Chen F, Ren W, Lin Z (2010), Multi-agent coordination with cohesion, dispersion, and containment control, American Control Conference (ACC), pp. 4756-4761.

[13] Ziying Z, Rubo Z, Xin L (2008), Multi-robot formation control based on behaviour, International Conference on Computer Science and Software, vol. 2, pp. 1045-1048.

[14] Fidan B, Gazi V, Zhai S, Cen, N, Karatas E (2012), Single view distance estimation based formation control of robotic swarms, IEEE Transactions on Industrial Electronics, p. 1.

[15] Chang B. L (2011), A dynamic virtual structure formation control for fixed-wing UAVs, 9th IEEE International Conference on Control and Automation pp. 627-632.

[16] Zhao W, Tiauw H. G (2010), Robust cooperative leader-follower formation flight control, 11th International Conference on Control Automation Robotics & Vision, pp. 275-280.

[17] Kowdiki K. H, Barai R. K, Bhattacharya S (2012), Leader-follower formation control using artificial potential functions: A kinematic approach, International Conference on Advances in Engineering, Science and Management, pp. 500-505.

[18] Dong W (2012), Tracking control of multiple-wheeled mobile robots with limited information of a desired trajectory, IEEE Transactions on Robotics, vol. 28, No.1, pp. 262-268.

[19] Savkin A. V, Chao W, Baranzadeh A, Zhiyu X (2013), Method for decentralized formation building for unicycle-like mobile robots, 9th Asian Control Conference (ASCC).

[20] Lin Z, Francis B, Maggiore M (2005), Necessary and sufficient graphical conditions for formation control of unicycles, IEEE Transactions on Automatic Control, vol. 50, No.1, pp. 121-127.

[21] Savkin A. V, Teimoori H (2010), Decentralized navigation of groups of wheeled mobile robots with limited communication, IEEE Transactions on Robotics, vol. 26, No. 10, pp. 1099-1104.

[22] Hernandez-Martinez E. G, Flores Albino J. M (2011), Hybrid architecture of multi-robot systems based on formation control and SOM neural networks, IEEE International Conference on Control Applications, pp. 941-946.

[23] Instrument Society of America (1999), ISA-95.1. Enterprise-control system integration, Part 1. Models and terminology, ISA Standards.

[24] Gradišar D, Mušič G (2012), Petri Nets – Manufacturing and Computer Science, edited by P. Pawlewski, InTech DOI,10.5772/48467 pp. 5-26.

[25] Sanchez A, Aranda-Bricaire E, Jaimes F, Hernandez E, Nava A (2010), Synthesis of product-driven coordination controllers for a class of discrete-event manufacturing systems, Robotics and Computer-Integrated Manufacturing, vol. 26, No. 4, pp. 361-369.

[26] Celaya J. R, Desrochers A, Graves R. J (2009), Modeling and analysis of multi-agent systems using Petri nets, Journal of Computers, vol. 4, No. 10, pp. 981-996.

[27] Mahfoudhi A, Marzougui B, Abid M (2010), Agents Petri nets: Theory and application, International Journal of Sciences and Techniques of Automatic Control & Computer Engineering, vol. 4, No. 2, pp. 1402-1419.

[28] Marzougui B, Kassine L, Kamel B, (2010), A new formalism for modeling Multi Agent Systems: Agent Petri nets, J. Software Engineering & Applications pp. 1118-1124.

[29] Sanchez A, Aranda-Bricaire E, Hernandez-Martinez E.G, Magallon J, Molina J (2009), AGV navigation in flexible manufacturing systems using formation control. Congreso Anual de la Asociación de México de Control Automático (AMCA).

[30] Hernández-Martínez E. G, Aranda-Bricaire E (2011), Convergence and collision avoidance in formation control: A survey of the artificial potential functions approach, multi-agent systems – Modeling, control, programming, simulations and applications, edited by F. Alkhateeb, InTech, DOI: 10.5772/14142.

[31] Desai J. P, Ostrowski J. P, Kumar V (2001), Modeling and control of formations of nonholonomic mobile robots, IEEE Transactions on Robotics and Automation, vol. 17, No. 6, , pp. 905-908.

[32] Brockett R. W (1983), Asymptotic stability and feedback stabilization, in Differential Geometric Control Theory, edited by R. Brockett, R. S. Millman, H. J. Sussmann, Boston, Birkhauser, pp. 181-191.

[33] Hernández-Martínez E.G, Aranda-Bricaire E (2010), Trajectory tracking for groups of unicycles with convergence of the orientation angles, Proceedings of the 49th IEEE Conference on Decision and Control, pp. 6323-6328.